A brief comparison of Simon and Simeck

Kölbl, Stefan; Roy, Arnab

Published in:
Lecture Notes in Computer Science

Link to article, DOI:
10.1007/978-3-319-55714-4_6

Publication date:
2017

Document Version
Peer reviewed version

A Brief Comparison of Simon and Simeck

Stefan Kölbl, Arnab Roy
{stek,arroy}@dtu.dk

DTU Compute, Technical University of Denmark, Denmark

Abstract. Simeck is a new lightweight block cipher design based on combining the design principles of the Simon and Speck block cipher. While the design allows a smaller and more efficient hardware implementation, its security margins are not well understood. The lack of design rationals of its predecessors further leaves some uncertainty on the security of Simeck.
In this work we give a short analysis of the impact of the design changes by comparing the upper bounds on the probability of differential and linear trails with Simon. We also give a comparison of the effort of finding those bounds, which surprisingly is significantly lower for Simeck while covering a larger number of rounds at the same time.
Furthermore, we provide new differentials for Simeck which can cover more rounds compared to previous results on Simon and study how to choose good differentials for attacks and show that one can find better differentials by building them from a larger set of trail with initially lower probability.
We also provide experimental results for the differentials for Simon32 and Simeck32 which show that there exist keys for which the probability of the differential is significantly higher than expected.
Based on this we mount key recovery attacks on 19/26/33 rounds of Simeck32/48/64, which also give insights on the reduced key guessing effort due to the different set of rotation constants.

Keywords: SIMON, SIMECK, differential cryptanalysis, block cipher

1 Introduction

Simeck is a family of lightweight block ciphers proposed in CHES'15 by Yang, Zhu, Suder, Aagaard and Gong [13]. The design combines the Simon and Speck block ciphers proposed by NSA [4], which leads to a more compact and efficient implementation in hardware. The block cipher Simon is built by iterating a very simple round function which uses bitwise AND and rotation while the block cipher Speck uses modular addition as non-linear operations. The designers of Simeck chose a different set of rotation constants from Simon to construct the round function.

The efficiency of Simon and Speck on hardware and software platform has a natural appeal to use similar design principles for constructing efficient primitives. The designers of Simon and Speck do not provide rationales for the original choices apart from implementation aspects. These modifications are likely

to have an impact on the security margins, which often are already small for lightweight designs and can be a delicate issue. Hence it is important to understand the effect of the parameter change on the security of SIMON like design.

The SIMON block cipher family has been studied in various paper [1, 2, 5, 9, 10, 12] and the attacks covering the most rounds are based on differential and linear cryptanalysis, which therefore will also be the focus of this work. However very few analyses [7] was done to study the choice of parameters for SIMON and SPECK and their effect on the security of these block ciphers.

Our Results In this paper we give a first analysis on the impact of these design changes by comparing the bounds for differential and linear trails with the corresponding variants of SIMON. An unexpected advantage for SIMECK is, that it takes significantly less time to find those while also covering more rounds (see Table 1). Additionally we investigate strategies to find differentials which have a high probability and are more suitable for efficient attacks.

Surprisingly, we can find differentials with higher probability for SIMECK32 by not using the input and output difference from the best differential trails. Furthermore, we also provide new differentials which cover 4 and 5 additional rounds for SIMECK48 and SIMECK64 respectively which also have a slightly higher probability compared to previous results on SIMON.

We verified the estimated probability with experiments for both SIMON32 and SIMECK32 to confirm our model and also noticed that for some keys a surprisingly large number of valid pairs can be found.

This is followed by key-recovery attacks for reduced round versions of SIMECK (see Table 6). These attacks are similar to previous work [5] done on SIMON and give insight into the lower complexity for the key recovery process for SIMECK as we need to guess fewer key bits.

Table 1: A comparison between the number of rounds for which upper bounds on the probability of differential and linear trails exist, the probability of differentials utilized in attacks and the best differential attacks on SIMON and SIMECK. Results contributed by this work are marked in bold.

Cipher	Rounds	Upper Bounds		Differentials		Key Recovery
		differential	linear	Rounds	$\Pr(\alpha \to \beta)$	
SIMON32/64	32	**32**	**32**	13	$2^{-28.79}$ [5]	21 [11]
SIMECK32/64	32	**32**	**32**	**13**	$\mathbf{2^{-27.28}}$	22 [8]
SIMON48/96	36	**19**	**20**	16	$2^{-44.65}$ [10]	24 [11]
SIMECK48/96	36	**36**	**36**	**20**	$\mathbf{2^{-43.65}}$	26 [8]
SIMON64/128	44	15 [7]	**17**	21	$2^{-60.21}$ [10]	29 [11]
SIMECK64/128	44	**40**	**41**	**26**	$\mathbf{2^{-60.02}}$	35 [8]

2 The Simeck Block Cipher

SIMECK2n is a family of block ciphers with n-bit word size, where $n = 16, 24, 32$. Each variant has a block size of $2n$ and key size of $4n$ giving the three variants of SIMECK: SIMECK32/64, SIMECK48/96 and SIMECK64/128. As for each block size there is only one key size we will omit the key size usually.

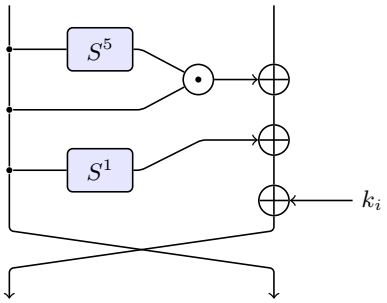


Fig. 1: The round function of SIMECK.

The block cipher is based on the *Feistel* construction and the round function f is the same as in SIMON apart from using $(5, 0, 1)$ for the rotation constants (as depicted in Figure 1). The key-schedule on the other hand is similar to SPECK, reusing the round function to update the keys. The key K is split into four words (t_2, t_1, t_0, k_0) and the round keys $k_0, \ldots, k_{r-1}$ are given by:

$$\begin{aligned} k_{i+1} &= t_i \\ t_{i+3} &= k_i \oplus f(t_i) \oplus C \end{aligned} \tag{1}$$

3 Preliminaries

Differential cryptanalysis is a powerful tool for analyzing block ciphers using a chosen plaintext attack. The idea is to find a correlation between the difference of a pair of plaintexts and the corresponding pair of ciphertexts. Resistance to differential cryptanalysis is an important design criteria but it is difficult, especially for designs like SIMON, to proof the resistance against it.

Definition 1. *A* differential trail Q *is a sequence of difference patterns*

$$Q = (\alpha_0 \xrightarrow{f_0} \alpha_1 \xrightarrow{f_1} \cdots \alpha_{r-1} \xrightarrow{f_{r-1}} \alpha_r). \tag{2}$$

In general, as the key is unknown to an attacker, we are interested in the probability that a random pair of inputs follows such a differential trail and the goal for the attacker is to find a correlation between input and output difference with high probability.

Definition 2. *The probability of a differential trail Q is defined as*

$$\Pr(\alpha_0 \xrightarrow{f_0} \alpha_1 \xrightarrow{f_1} \cdots \alpha_{r-1} \xrightarrow{f_{r-1}} \alpha_r) = \prod_{t=0}^{r-1} \Pr(\alpha_t \to \alpha_{t+1}) \tag{3}$$

and gives the probability that a random input follows the differential trail. The last equality holds if we assume independent rounds.

In most attack scenarios we are not interested in the probability of a differential trail, as we are only interested in the input difference α_0 and the output difference α_r, but not what happens in between.

Definition 3. *The probability of a* differential *is the sum of all r round differential trails*

$$\Pr(\alpha_0 \xrightarrow{f} \alpha_r) = \sum_{\alpha_1,\ldots,\alpha_{r-1}} (\alpha_0 \xrightarrow{f_0} \alpha_1 \xrightarrow{f_1} \cdots \alpha_{r-1} \xrightarrow{f_{r-1}} \alpha_r) \tag{4}$$

which have the same input and output difference.

4 Analysis of Simon and Simeck

In [7] the differential and linear properties of Simon were studied, including variants using a different set of rotation constants. Following up on this work, we can use the same methods to analyze the round function of Simeck. This allows us to find lower bounds for the probability of a differential trail resp. square correlation of a linear trail for a given number of rounds.

4.1 Diffusion

An important criteria for the quality of a round function in a block cipher is the amount of diffusion it provides, i.e. how many rounds r it takes until each bit at the input effects all bits of the output. For Simon this was already studied in [7] for the whole parameter set and we only explicitly state the comparison to Simeck here in Table 2.

Table 2: Number of rounds required for full diffusion.

Wordsize	32-bit	48-bit	64-bit
Simon	7 Rounds	8 Rounds	9 Rounds
Simeck	8 Rounds	9 Rounds	11 Rounds

4.2 Bounds on the best differential trails

We carried out experiments for the parameter set of SIMECK using CryptoSMT[1] to find the optimal differential and linear trails for SIMECK32, SIMECK48 and SIMECK64 and compare it with the results on SIMON. The results of this experiment are given in Figure 2. The bounds on the square correlation for linear trails are given in the Appendix.

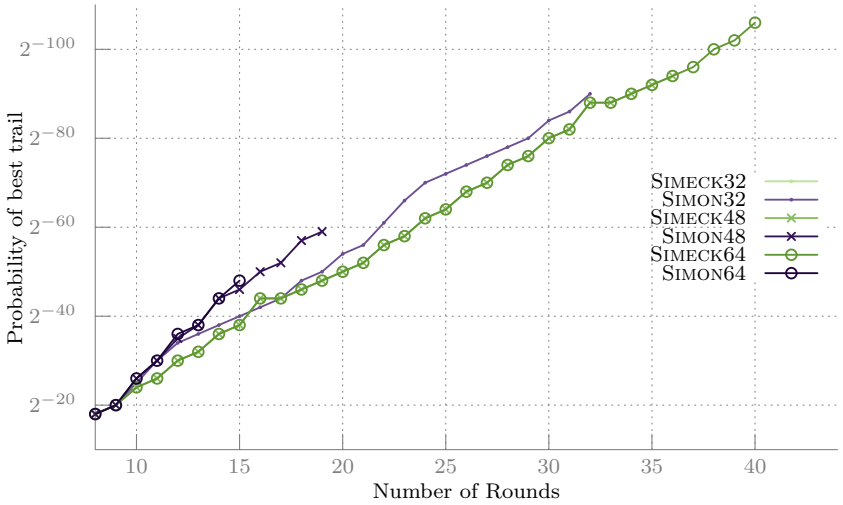


Fig. 2: Lower bounds on the probability of the best differential trails for variants of SIMON and SIMECK. For the different variants of SIMECK the bounds are the same.

While the bounds for SIMON32 and SIMECK32 are still comparable we noticed a significant difference for the larger variants. While the required number of rounds for SIMON48, such that the probability of the best trail is less than 2^{-48}, is 16, SIMECK48 achieves the same property only after 20 rounds. It is also interesting to note that the bounds for the different word sizes of SIMECK are the same, which is not the case for SIMON.

In our experiments we noticed that the different set of rotation constants plays a huge role in the running time of the SMT solver. For instance finding the bounds in Figure 2 took 51 hours for SIMON32 and 10 hours for SIMECK32[2]. Especially for larger block sizes it allows us to provide bounds for a significant larger number of rounds including full SIMECK48. For SIMON64 computing the bounds up to 15 rounds takes around 19 hours, while the same process only

[1] CryptoSMT https://github.com/kste/cryptosmt Version: 70794d83

[2] Using Boolector 2.0.1. running on an Intel Xeon X5650 2.66GHz 48GB RAM (1 core).

takes around 30 minutes for SIMECK64. We computed the bounds for SIMECK64 up to round 40 in around 53 hours.

4.3 Differential effect in Simon and Simeck

As noted in previous works SIMON shows a strong differential resp. linear hull effect, which invalidates an often made assumption that the probability of the best trail can be used to estimate the probability of the best differential. Therefore bounds on differential and linear trails have to be treated with caution. The choice of constants for SIMON-like round functions also plays a role in this as shown in [7].

One approach to find good differentials is to first find the best trail for a given number of rounds of SIMECK using CryptoSMT [6] and then find a large set of trails with the same input and output difference. However, as we will see later this will not always give the highest probability differential. The results of these experiments are summarized in Table 3.

If we compare those with previous results on SIMON we can cover more rounds. The best previous differential attack by Wang, Wang, Jia and Zhao [11] utilizes a 13-round differential for SIMON32, a 16-round differential for SIMON48 and a 21-round differential for SIMON64. We show that with the same or slightly better probability (Table 1) differentials can be found for a higher number of rounds for both SIMECK48 and SIMECK64.

Table 3: Overview of the differentials we found for SIMECK which can likely be used to mount attacks. The probability is given by summing up all trails up to probability $2^{\max}$ taking a time T.

Cipher	Rounds	$Q = (\alpha \to \beta)$	$\log_2(p)$	max	T
SIMECK32	13	$(8000, 4011) \to (4000, 0)$	-27.28	-49	17h
SIMECK48	20	$(20000, 450000) \to (30000, 10000)$	-43.65	-98	135h
SIMECK48	20	$(400000, e00000) \to (400000, 200000)$	-43.65	-74	93h
SIMECK48	21	$(20000, 470000) \to (50000, 20000)$	-45.65	-100	130h
SIMECK64	25	$(2, 40000007) \to (40000045, 2)$	-56.78	-90	110h
SIMECK64	26	$(0, 4400000) \to (8800000, 400000)$	-60.02	-121	120h

While we let our experiments run for a few days, the probability only improves marginally after a short time. For instance, for SIMECK32 and SIMECK48 the estimates after three minutes are only 2^{-2} lower than the final results and after two hours the improvements are very small. Some additional details on the differential utilized in the key-recovery attack on SIMECK48 can be found in the Appendix 9, including the exact running times to obtain the results.

4.4 Choosing a good differential for attacks

For an attack we want a differential with a high probability, but also the form of the input and output difference can have an influence on the resulting attack complexity. Ideally we want differentials with a sparse input/output difference resp. of the form $(x, 0) \rightarrow (0, x)$. When expanding such a differential it leads to a truncated differential with fewer unknown bits which reduces the complexity in the key recovery part of the attack as will be seen later.

The best differential trail of the form $(x, 0) \rightarrow (0, x)$ only has a probability of 2^{-42} for SIMECK32 resp. 2^{-47} for SIMON32. The corresponding differential improves the probability to $\approx 2^{-36.7}$, but is still unlikely to be useful for an attack. If we relax the restriction and allow differentials of the form $(x, x) \rightarrow (0, x)$ we can find differential trails with a probability of 2^{-38} (the same bound exists for SIMON32). However, the corresponding differentials still seem impractical for an attack. As both this approaches fail for finding good differentials we do not impose any restrictions on the form of the input resp. output difference of the differentials.

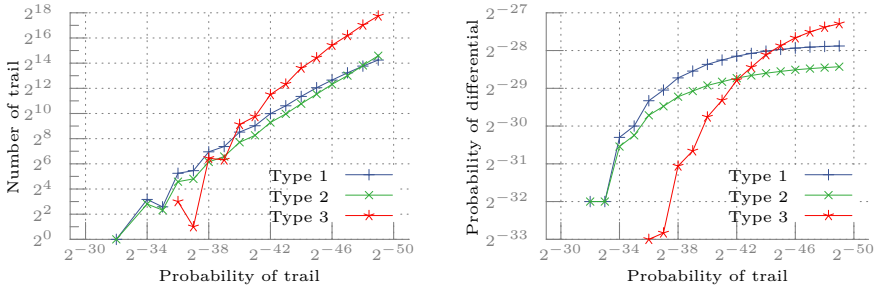
Table 4: Number of differential trails for 13-round SIMECK32.

$\Pr(\alpha \xrightarrow{f^{13}} \beta)$	Trails
2^{-32}	640
2^{-33}	128
2^{-34}	31616
2^{-35}	49152

We looked at all 40 rotation invariant differentials constructed from the best differential trail with probability 2^{-32} for SIMECK32 (see Table 4). There are only two possible distributions for the trails contributing to the differential, which we denote as Type 1 and Type 2 (see Figure 3 and Table 8). There are 8 trails of Type 1, all with at least one word having 0 difference, and the corresponding differential gives a slightly higher probability. For a list of these differentials see Table 7.

However, by expanding our search we could find a better differential. By not using the optimal differential trail we can find the differential $(8000, 4011) \rightarrow (4000, 0)$ which has a higher probability even though the best trail contributing only has a probability of 2^{-36}. This is due to the higher number of trails contributing to this specific differential (see Type 3 in Figure 3 respectively Table 8).

For 20-round SIMECK48 the best trails with pattern only has a probability of 2^{-62} and for $(x, x) \rightarrow (0, x)$ it is 2^{-54}. The corresponding differentials are not usable for an attack in this case. Therefore, we again do not impose any of these



(a) Number of trails (b) Accumulated Probability

Fig. 3: Distribution of trails contributing to the differentials for 13 rounds of SIMECK32 and the accumulated probability by summing up all trails up to a specific probability.

restrictions and use the 20-round trails with highest probability. For SIMECK48 there are 768 such trails with a probability of 2^{-50} (32 rotation invariant) and we choose the one where the input and output difference is most sparse.

For SIMECK64 the best differentials we found are also based on the best trail and given in Table 3.

4.5 Experimental Verification

While the previous approach can give a good estimate for the probability one can expect for a differential, it is not entirely clear how good these approximations are. As both SIMON32 and SIMECK32 allow us to run experiments on the full codebook we can verify the probabilities at least for these variants. For a random function we expect that the number of valid pairs are a Poisson distribution.

Definition 4. *Let X be a Poisson distributed random variable representing the number of pairs (a, b) with values in $\mathbb{F}_2^n$ following a differential $Q = (\alpha \xrightarrow{f} \beta)$, that means $f(a) \oplus f(a \oplus \alpha) = \beta$, then*

$$\Pr(X = l) = \frac{1}{2}(2^n p)^l \frac{e^{-(2^n p)}}{l!} \tag{5}$$

where p is the probability of the differential.

We ran experiments for both SIMON32 and SIMECK32 reduced to 13 rounds by encrypting the full code book for a large number of random keys. The differential we used for SIMON32 is $(0, 40) \rightarrow (4000, 0)$, which is also used in the best attack so far [11] and has an estimated probability of $2^{-28.56}$. The expected number of valid pairs is $\mathbf{E}(X) \approx 5.425$. We encrypted the full code book using 202225 random master keys and counted the number of unique pairs. The full distribution is given in Figure 4. The distribution follows the model in Equation 5, but

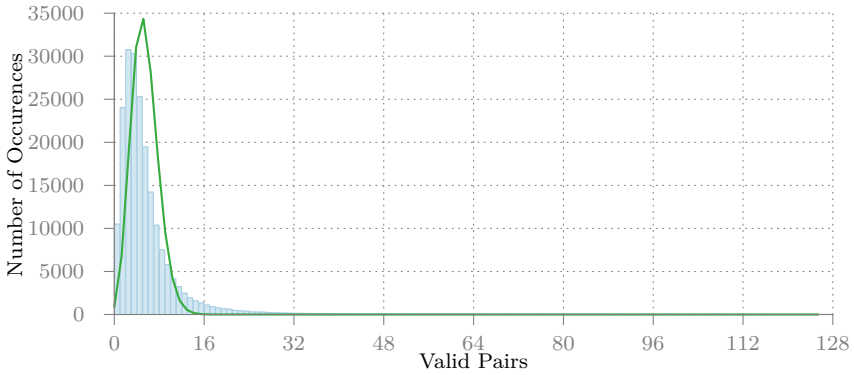


Fig. 4: Distribution of how many times we observe l valid pairs for the differential $(0, 40) \xrightarrow{f^{13}} (4000, 0)$ for SIMON32 using a random key.

we observe some unusual high number of pairs for some keys. For example the key $K = (k_0, k_1, k_2, k_3) = (\texttt{8ec1}, \texttt{1cf8}, \texttt{e84a}, \texttt{cee2})$ gives 1082 pairs following the differential. If 13 rounds of SIMON32 would behave like a random function, this would only occur with an extremely low probability $\Pr(X = 1082) \ll 2^{-1000}$.

For SIMECK32 we used the new differential $(8000, 4011) \rightarrow (4000, 0)$ with $\mathbf{E}(X) \approx 13.175$. Again, we encrypt the full code book for 134570 random keys and the distribution follows our model as can be seen in Figure 5. Similar, to SIMON for some keys a surprisingly large number of valid pairs can be found. In both cases our method provides a good estimate for the probability of a

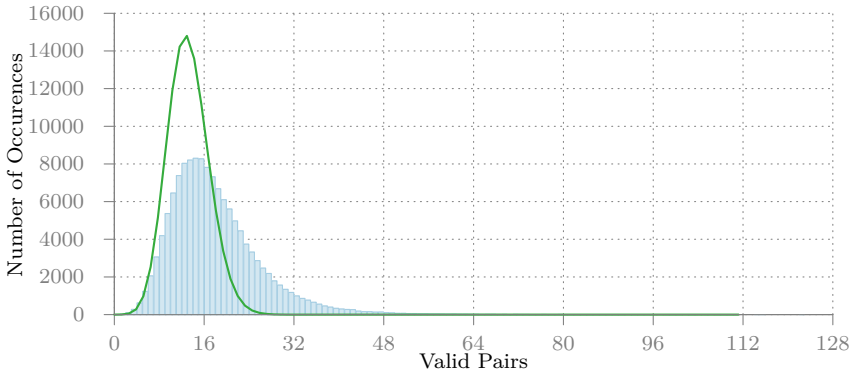


Fig. 5: Distribution of how many times we observe l valid pairs for the differential $(8000, 4011) \xrightarrow{f^{13}} (4000, 0)$ for SIMECK32 using a random key.

differential and we can use Equation 5 for estimating the number of pairs.

5 Recovering the Key

Table 5: Truncated differential obtained by extending $(400000, e00000) \xrightarrow{20} (400000, 200000)$ in both directions until all bits are unknown.

Round	ΔL	ΔR	$*$	$*$
-5	***0***0**********	**********	22	24
-4	***000000***0**********	***0***0**********	17	22
-3	***00000000000***0****1*	***000000***0**********	11	17
-2	***0000000000000000***01	***00000000000***0****1*	6	11
-1	11100000000000000000000	***0000000000000000***01	0	6
0	01000000000000000000000	11100000000000000000000	0	0
	20 rounds			
20	01000000000000000000000	00100000000000000000000	0	0
21	1*10000000000000000*000	01000000000000000000000	2	0
22	***000000000000*000***01	1*10000000000000000*000	7	2
23	***0000000*000***0****1*	***000000000000*000***01	12	7
24	***00*000***0**********	***0000000*000***0****1*	18	12
25	***0***0**********	***00*000***0**********	22	18
26	**********	***0***0**********	24	22

In the following subsection we describe the key recovery attack on SIMECK48 based on the differential given in Table 3. Extending this differential both in forward and backward directions gives the truncated differential shown in Table 5 which will be used in the attack. The input difference to round r is denoted as Δ^r and k_r denotes the round key for round r. The difference in the left resp. right part of the state we denote as ΔL^r and ΔR^r.

5.1 Attack on 26-round Simeck48

Our attack on 26-round SIMECK48 uses four 20-round differentials in a similar way as in [5]. Let D_i denote the differentials

$$\mathsf{D}_1 : (400000,\ e00000) \xrightarrow{f^{20}} (400000,\ 200000)$$
$$\mathsf{D}_2 : (800000,\ c00001) \xrightarrow{f^{20}} (800000,\ 400000)$$
$$\mathsf{D}_3 : (000004,\ 00000e) \xrightarrow{f^{20}} (000004,\ 000002)$$
$$\mathsf{D}_4 : (000008,\ 00001c) \xrightarrow{f^{20}} (000008,\ 000004)$$

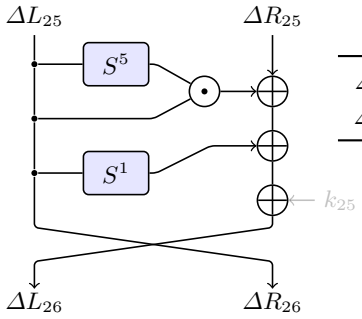
each having probability $\approx 2^{-44}$. We add 4 rounds at the end and 2 rounds on top and obtain the truncated difference (see Table 5). The truncated difference at round 0 for each differential is given by

$$***00000000000000000***01, ***000000000000***0****1*$$
$$**000000000000000***01*, **000000000000***0****1**$$
$$000000000000000***01***0, 000000000000***0****1****0$$
$$000000000000000***01***00, 000000000000***0****1****00\ .$$

By identifying the unknown and known bit positions in these differentials we can construct a set of 2^{30} plaintext pairs where the bit positions corresponding to the aligned 0s in the truncated differentials are fixed to an arbitrary value for all plain-texts. By guessing 6 round key bits we can also identify the 2^{31} pairs satisfying the difference $(\Delta L^2, \Delta R^2)$ after the first two round encryption. Hence we can get 4 sets of 2^{31} pairs of plain-texts where the difference is satisfied after the first two rounds of encryption. By varying the fixed bit positions we can get 4 sets of 2^{46} pairs of plain-texts, each satisfying the difference after two rounds for each key guess.

Filtering the pairs First we encrypt the 2^{46} plaintext pairs. Then we unroll the last round and use the truncated differential to verify if a pair is valid. This is possible due to the last key addition not having any influence on the difference $(\Delta L^{25}, \Delta R^{25})$. As there are $12 + 17$ bits known in this round we will have $2^{46-29} = 2^{17}$ plaintext pairs left.

Key guessing In the key guessing phase we guess the necessary round key bits (or linear combination of round key bits) to verify the difference at the beginning of round 22, i.e. Δ^{22}. For each differential we counted that a total of 30 round



ΔL_{25}	***0000000*000***0****1*	12
ΔR_{25}	***000000000000*000***01	17

For each pair $(\Delta L', \Delta R')$ obtained:

1. Compute $\Delta R'_{25} = f(\Delta R'_{26}) \oplus \Delta L_{26}$[3].
2. Check if $\Delta L'_{25} = \Delta L_{25}$ and $\Delta R'_{25} = \Delta R_{25}$.

Fig. 6: Filtering for the correct pairs which we use in the key guessing part.

key bits and linear combinations of round key bits are necessary to be guessed during this process. The required key bits $\mathsf{D}_1^{\mathsf{K}}$ for D_1 are

$$K^{23} = \{2, 17\}$$
$$K^{24} = \{2, 3, 4, 8, 12, 16, 17, 18, 22\}$$
$$K^{25} = \{1, 2, 3, 4, 5, 7, 8, 9, 11, 12, 13, 15, 16, 17, 18, 19, 21, 22, 23\}$$

We describe this process for one round in Figure 7. An interesting difference to SIMON in the key guessing part is that the required number of key guesses is much lower, as many bits required to guess coincide when partially recovering the state which can reduce the overall complexity. This is always the case if one of the rotation constants is zero, but similar effects can occur with other choices as well.

For the key guessing part, we keep an array of 2^{30} counters and increment a counter when it is correctly verified with the difference after partial decryption of the cipher-text pairs. For each differential we can verify the remaining $19 (= 48 - 29)$ bits with the key guessing process. For the 2^{30} counters we expect to have $(2^{17} \times 2^{30})/2^{19} = 2^{28}$ increments. The probability of a counter being incremented is $2^{28}/2^{30} = 2^{-2}$. Since 4 correct pairs are expected to be among the filtered pairs, the expected number of counters having having at least 4 increments is

$$2^{30} \cdot (1 - \Pr(X < 4)) \approx 2^{17.13}. \tag{6}$$

We observe that there are 18 common key guesses required for the differentials D_1 and D_2. Hence combining the corresponding array of counters T_1 and T_2 we can get $2^{17.13} \times 2^{17.13}/2^{18} = 2^{16.26}$ candidates for 42 bits. Continuing in the same way we observe that $|\mathsf{D}_3^K \cap (\mathsf{D}_1^K \cup \mathsf{D}_2^K)| = 24$, hence we get $2^{16.26} \times 2^{17.13}/2^{24} = 2^{9.39}$ candidates for 48 bits. Using D_4 this can be further reduced, as $|\mathsf{D}_4^K \cap (\mathsf{D}_1^K \cup \mathsf{D}_2^K \cup \mathsf{D}_3^K)| = 28$ we expect $2^{9.39} \times 2^{17.13}/2^{28} \approx 2^{-1.5}$ candidates for 50 bits. For the remaining 46 bits we perform an exhaustive search.

[3] The key has no influence on the input to the non-linear function in the last round.

Complexity The complexity of the attack is dominated by the key recovery process. For the partial decryption process we need $2^{17} \times 2^{30} \times \frac{4}{26} \approx 2^{45}$ encryptions, hence the complexity of one key recovery attack is 2^{54}. This key recovery is performed for each differential and each 2^6 round key guesses of the initial rounds. Hence the overall complexity of the attack is $2^{54} \times 2^6 \times 4 = 2^{62}$.

We expect in our attack that at least 4 out of 2^{46} pairs follow our differential, which has probablity $\geq 2^{-43.65}$, for the correct key. Therefore we get a success rate of

$$1 - \Pr(X < 4) \approx 0.75 \tag{7}$$

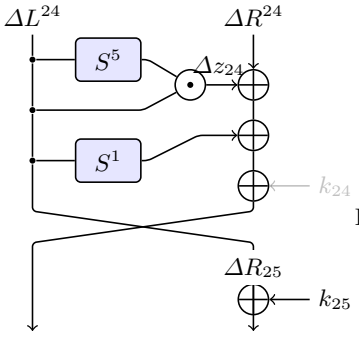
However, in practice this will be much higher as we only use a lower bound on the probability of the differential.

5.2 Key Recovery for 19-round Simeck32

For SIMECK32 we also use 4 differentials

$$\begin{aligned}
\mathsf{D}_1 &: \texttt{(8000, 4011)} \xrightarrow{f^{13}} \texttt{(4000, 0000)} \\
\mathsf{D}_2 &: \texttt{(0001, 8022)} \xrightarrow{f^{13}} \texttt{(8000, 0000)} \\
\mathsf{D}_3 &: \texttt{(0008, 0114)} \xrightarrow{f^{13}} \texttt{(0004, 0000)} \\
\mathsf{D}_4 &: \texttt{(0010, 0228)} \xrightarrow{f^{13}} \texttt{(0008, 0000)}
\end{aligned}$$

each having probability $\approx 2^{-28}$ (for the truncated differences see Table 10). We add two rounds at the top of the 13-round differential and identify a set of 2^{30} pairs of plain-texts each satisfying the specific difference $(\Delta L^2, \Delta R^2)$ after the first two round encryption. Identifying a set of plaintext pairs requires to guess 6 key bits.



ΔR_{25}	∗∗∗0000000000000∗0000∗∗∗01
$S^5(\Delta R_{25})$	0000000000∗000∗∗∗01∗∗∗00
Δz_{24}	∗∗∗0000000∗000∗∗∗0∗∗∗∗0∗
ΔR_{24}	1∗1000000000000000000∗000

Key filtering:

1. Find bits s.t. $\Delta z_{24} = *$ and $\Delta R_{24} \neq *$.
2. Guess corresponding bits in k_{25}.
3. Check $\Delta z_{24} = \Delta R_{24} \oplus S^1(R_{25}) \oplus \Delta L_{25}$.

Fig. 7: Outline of the process of key guessing and filtering for a single round.

Filtering We can filter some wrong pairs by unrolling the last round and verifying the truncated difference (with 18 known bits) at the beginning of the last round. This will leave us with $2^{30-18} = 2^{12}$ pairs.

Key guessing We counted that 22 round key bits are necessary to guess for verifying the difference at the end of round 14. The required key bits $\mathsf{D}_1^{\mathsf{K}}$ for D_1 are

$$K^{16} = \{3, 9\}$$
$$K^{17} = \{2, 3, 4, 8, 9, 10, 14\}$$
$$K^{18} = \{1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 13, 14, 15\}$$

We use the same method as described for SIMECK48 during this phase. Out of the filtered pairs we expect to get at least 4 correct pairs (those follow the 13-round differential). Hence the number of candidates for 22 key bits are $\approx 2^{9.1}$. The number of common key bits amongst the differentials is given by

$$\mathsf{D}_1^{\mathsf{K}} \cap \mathsf{D}_2^{\mathsf{K}} = 14$$
$$\mathsf{D}_3^{\mathsf{K}} \cap (\mathsf{D}_1^{\mathsf{K}} \cup \mathsf{D}_2^{\mathsf{K}}) = 16$$
$$\mathsf{D}_4^{\mathsf{K}} \cap (\mathsf{D}_1^{\mathsf{K}} \cup \mathsf{D}_2^{\mathsf{K}} \cup \mathsf{D}_3^{\mathsf{K}}) = 20$$

and we expect to 1 key candidate for 38 bits. For the remaining 26 bits of the last four round keys we perform exhaustive search.

Complexity The complexity of the partial decryption (for the last 4 rounds) is $2^{12} \times 2^{22} \times \frac{4}{19} \approx 2^{32}$ which is the dominating part of the complexity. Since we perform the key recovery for each differential and for each 6-bit round key guesses of the first two rounds the overall complexity of the attack is $2^{32+8} = 2^{40}$.

5.3 Key Recovery for 33-round Simeck64

We use the following 4 differentials for SIMECK64

$$\mathsf{D}_1 : \texttt{(0, 04400000)} \xrightarrow{f^{26}} \texttt{(08800000, 00400000)}$$
$$\mathsf{D}_2 : \texttt{(0, 44000000)} \xrightarrow{f^{26}} \texttt{(88000000, 04000000)}$$
$$\mathsf{D}_3 : \texttt{(0, 40000004)} \xrightarrow{f^{26}} \texttt{(80000008, 40000000)}$$
$$\mathsf{D}_4 : \texttt{(0, 00000044)} \xrightarrow{f^{26}} \texttt{(00000088, 00000004)}$$

each having probability $\approx 2^{-60}$ (for the truncated differences see Table 11). We add two rounds at the top of the 26 round differential and identify a set of 2^{62} pairs of plain-texts by guessing 4 round key bits from the first two rounds.

Filtering wrong pairs We add 5 round truncated difference at the end of the 26 round differential. The last round may be unrolled to verify the difference at the beginning of the last round. This helps to filter some wrong pairs using the known bits of the truncated difference and after filtering we are left with $2^{62-30} = 2^{32}$ pairs of plaintext out of which we expect 2^2 correct pairs (those followed 26 round differential).

Key guessing In this phase we guess the necessary key bits from the last four rounds to verify the difference at the beginning of round 28. We counted that 76 key bits are necessary to guess for verifying $(\Delta L^{28}, \Delta R^{28})$. The required key bits $\mathsf{D}_1^{\mathsf{K}}$ for D_1 are

$$\begin{aligned} K^{29} &= \{0, 18, 22, 28\} \\ K^{30} &= \{0, 1, 5, 13, 17, 18, 19, 21, 22, 23, 27, 28, 29, 31\} \\ K^{31} &= \{0, 1, 2, 4-6, 8, 10, 12-14, 16-24, 26-31\} \\ K^{32} &= \{0-31\} \end{aligned}$$

Out of the filtered pairs we expect to get at least 4 correct pairs (those that follow the 26-round differential). Hence the number of candidates for 76 key bits are $\approx 2^{63.12}$. The number of common key bits amongst the differentials is given by

$$\begin{aligned} \mathsf{D}_1^{\mathsf{K}} \cap \mathsf{D}_2^{\mathsf{K}} &= 66 \\ \mathsf{D}_3^{\mathsf{K}} \cap (\mathsf{D}_1^{\mathsf{K}} \cup \mathsf{D}_2^{\mathsf{K}}) &= 70 \\ \mathsf{D}_4^{\mathsf{K}} \cap (\mathsf{D}_1^{\mathsf{K}} \cup \mathsf{D}_2^{\mathsf{K}} \cup \mathsf{D}_3^{\mathsf{K}}) &= 64 \end{aligned}$$

By combining all the four differentials we expect to get 2^{52} key candidates for 104 bits. For the remaining 24 bits of the last four round keys we perform exhaustive search.

Complexity The complexity of the partial decryption (for last 4 rounds) is $2^{32} \times 2^{76} \times \frac{5}{33} \approx 2^{105}$ which is the dominating part of the complexity. Since we perform the key recovery for each differential and for each 6-bit round key guesses of the first two rounds the overall complexity of the attack is $2^{105+10} = 2^{115}$.

6 Conclusion and Future Work

We gave a brief overview of the SIMECK and SIMON block cipher and their resistance against differential and linear cryptanalysis. From our comparison we can see that statistical attacks can cover a significant larger number of rounds for SIMECK48 and SIMECK64. Our key recovery attacks still have a significant margin compared to generic attacks (see Table 6) in regard to time complexity, therefore additional rounds can be covered using the dynamic key-guessing approach at the costs of a higher complexity.

Table 6: Comparison of the attacks on SIMECK.

Cipher	Rounds	Time	Data	Memory	Type
SIMECK32/64	20/32	$2^{62.6}$	2^{32}	2^{56}	Imp. Differential [13]
SIMECK32/64	22/32	$2^{57.9}$	2^{32}	–	Diff.(dynamic key-guessing) [8]
SIMECK32/64	18/32	$2^{63.5}$	2^{31}	–	Linear [3]
SIMECK32/64	19/32	$\mathbf{2^{40}}$	$\mathbf{2^{31}}$	$\mathbf{2^{31}}$	Differential (Section 5.2)
SIMECK48/96	24/36	$2^{94.7}$	2^{48}	2^{74}	Imp. Differential [13]
SIMECK48/96	28/36	$2^{68.3}$	2^{46}	–	Diff.(dynamic key-guessing) [8]
SIMECK48/96	24/36	2^{94}	2^{45}	–	Linear [3]
SIMECK48/96	26/36	$\mathbf{2^{62}}$	$\mathbf{2^{47}}$	$\mathbf{2^{47}}$	Differential (Section 5.1)
SIMECK64/128	25/44	$2^{126.6}$	2^{64}	2^{79}	Imp. Differential [13]
SIMECK64/128	34/44	$2^{116.3}$	2^{63}	–	Diff.(dynamic key-guessing) [8]
SIMECK64/128	35/44	$2^{116.3}$	2^{63}	–	Diff.(dynamic key-guessing) [8]
SIMECK64/128	27/44	$2^{120.5}$	2^{61}	–	Linear [3]
SIMECK64/128	**33/44**	$\mathbf{2^{115}}$	$\mathbf{2^{63}}$	$\mathbf{2^{63}}$	Differential (Section 5.3)

This also shows that the impact of small design changes in SIMON-like block ciphers can be hard to estimate and requires a dedicated analysis, as the underlying design strategy is still not well understood. Especially for variants with a larger block size it is difficult to find lower bounds or estimate the effect of differentials. An open question is whether better differentials exist for both SIMON and SIMECK which give a surprisingly higher probability as in the case of our differential for SIMECK32. This effect could be more significant for larger word sizes and lead to improved attacks.

In this sense SIMECK also has an unexpected advantage over SIMON and SPECK, as the analysis is simpler and requires less computational effort with our approach. This is a property that is especially important in the light of not having cryptanalytic design documentation, nor design rationales for the constants regarding security available by the designers of SIMON and SPECK.

For both SIMON32 and SIMECK32 reduced to 13 rounds we observed that for some keys a surprisingly large number of valid pairs can be found. This gives an interesting open problem in classifying the keys which give a significant higher probability for a given differential.

Acknowledgments

We would like to thank the anonymous reviewers who helped improve the quality of this paper The work in this paper was in part supported by European Union's H2020 Programme under grant agreement number ICT-645622 (PQCRYPTO)

References

1. Abed, F., List, E., Lucks, S., Wenzel, J.: Differential cryptanalysis of round-reduced SIMON and SPECK. In: Cid, C., Rechberger, C. (eds.) Fast Software Encryption, FSE 2014. Lecture Notes in Computer Science, vol. 8540, pp. 525–545. Springer (2015)
2. Alizadeh, J., AlKhzaimi, H., Aref, M.R., Bagheri, N., Gauravaram, P., Kumar, A., Lauridsen, M.M., Sanadhya, S.K.: Cryptanalysis of SIMON variants with connections. In: Saxena, N., Sadeghi, A. (eds.) Radio Frequency Identification: Security and Privacy Issues, RFIDSec 2014. Lecture Notes in Computer Science, vol. 8651, pp. 90–107. Springer (2014)
3. Bagheri, N.: Linear cryptanalysis of reduced-round SIMECK variants. In: Progress in Cryptology - INDOCRYPT 2015. pp. 140–152 (2015)
4. Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., Wingers, L.: The SIMON and SPECK families of lightweight block ciphers. Cryptology ePrint Archive, Report 2013/404 (2013), http://eprint.iacr.org/
5. Biryukov, A., Roy, A., Velichkov, V.: Differential analysis of block ciphers SIMON and SPECK. In: Cid, C., Rechberger, C. (eds.) Fast Software Encryption, FSE 2014. Lecture Notes in Computer Science, vol. 8540, pp. 546–570. Springer (2015)
6. Kölbl, S.: CryptoSMT: An easy to use tool for cryptanalysis of symmetric primitives (2015), https://github.com/kste/cryptosmt
7. Kölbl, S., Leander, G., Tiessen, T.: Observations on the SIMON block cipher family. In: Advances in Cryptology - CRYPTO 2015. pp. 161–185 (2015)
8. Qiao, K., Hu, L., Sun, S.: Differential security evaluation of simeck with dynamic key-guessing techniques. Cryptology ePrint Archive, Report 2015/902 (2015), http://eprint.iacr.org/
9. Sun, S., Hu, L., Wang, M., Wang, P., Qiao, K., Ma, X., Shi, D., Song, L., Fu, K.: Constructing mixed-integer programming models whose feasible region is exactly the set of all valid differential characteristics of SIMON. Cryptology ePrint Archive, Report 2015/122 (2015), http://eprint.iacr.org/
10. Sun, S., Hu, L., Wang, P., Qiao, K., Ma, X., Song, L.: Automatic security evaluation and (related-key) differential characteristic search: Application to SIMON, PRESENT, LBlock, DES(L) and other bit-oriented block ciphers. In: Sarkar, P., Iwata, T. (eds.) Advances in Cryptology - ASIACRYPT 2014. Lecture Notes in Computer Science, vol. 8873, pp. 158–178. Springer (2014)
11. Wang, N., Wang, X., Jia, K., Zhao, J.: Differential attacks on reduced simon versions with dynamic key-guessing techniques. Cryptology ePrint Archive, Report 2014/448 (2014), http://eprint.iacr.org/
12. Wang, Q., Liu, Z., Varici, K., Sasaki, Y., Rijmen, V., Todo, Y.: Cryptanalysis of reduced-round SIMON32 and SIMON48. In: Meier, W., Mukhopadhyay, D. (eds.) Progress in Cryptology - INDOCRYPT 2014. Lecture Notes in Computer Science, vol. 8885, pp. 143–160. Springer (2014)
13. Yang, G., Zhu, B., Suder, V., Aagaard, M.D., Gong, G.: The simeck family of lightweight block ciphers. In: Cryptographic Hardware and Embedded Systems - CHES 2015. Springer (2015), to appear.

A Bounds for Linear trails

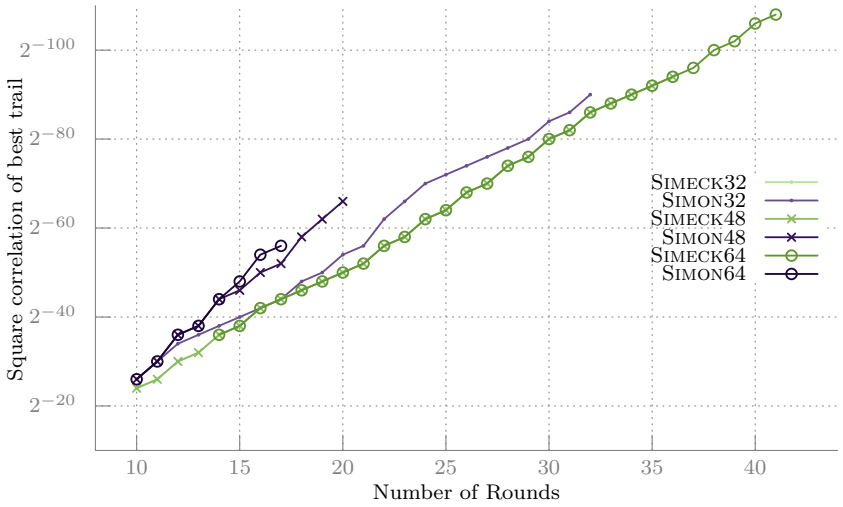


Fig. 8: Bounds for the best linear trails for variants of SIMON and SIMECK. For the different variants of SIMECK the bounds are the same.

Table 7: Classification of all the 40 rotation invariant 13-round differentials for SIMECK32.

Type 1			
$(0,22) \xrightarrow{f^{13}} (2a,1)$	$(4,8a8) \xrightarrow{f^{13}} (88,0)$	$(4,8e8) \xrightarrow{f^{13}} (88,)$	$(0,11) \xrightarrow{f^{13}} (1d,8)$
$(0,11) \xrightarrow{f^{13}} (115,8)$	$(0,88) \xrightarrow{f^{13}} (8e8,4)$	$(4,a8) \xrightarrow{f^{13}} (88,0)$	$(1,3a) \xrightarrow{f^{13}} (22,0)$
Type 2			
$(4,8a) \xrightarrow{f^{13}} (aa,4)$	$(4,8a) \xrightarrow{f^{13}} (ae,4)$	$(1,a8) \xrightarrow{f^{13}} (228,1)$	$(4,aa) \xrightarrow{f^{13}} (a,4)$
$(4,8e) \xrightarrow{f^{13}} (aa,4)$	$(4,2e) \xrightarrow{f^{13}} (a,4)$	$(4,2e) \xrightarrow{f^{13}} (e,4)$	$(2,57) \xrightarrow{f^{13}} (5,2)$
$(2,5) \xrightarrow{f^{13}} (55,2)$	$(4,8e) \xrightarrow{f^{13}} (2a,4)$	$(1,2a8) \xrightarrow{f^{13}} (228,1)$	$(2,7) \xrightarrow{f^{13}} (55,2)$
$(4,aa) \xrightarrow{f^{13}} (8e,4)$	$(4,ae) \xrightarrow{f^{13}} (e,4)$	$(4,8a) \xrightarrow{f^{13}} (2e,4)$	$(2,15) \xrightarrow{f^{13}} (5,2)$
$(2,7) \xrightarrow{f^{13}} (17,2)$	$(4,e) \xrightarrow{f^{13}} (ae,4)$	$(4,ae) \xrightarrow{f^{13}} (8e,4)$	$(4,8a) \xrightarrow{f^{13}} (2a,4)$
$(4,e) \xrightarrow{f^{13}} (2a,4)$	$(4,a) \xrightarrow{f^{13}} (2a,4)$	$(4,2e) \xrightarrow{f^{13}} (8a,4)$	$(4,2a) \xrightarrow{f^{13}} (8e,4)$
$(4,a) \xrightarrow{f^{13}} (ae,4)$	$(4,8e) \xrightarrow{f^{13}} (ae,4)$	$(1,28) \xrightarrow{f^{13}} (b8,1)$	$(4,8e) \xrightarrow{f^{13}} (2e,4)$
$(1,b8) \xrightarrow{f^{13}} (238,1)$	$(4,ae) \xrightarrow{f^{13}} (8a,4)$	$(2,15) \xrightarrow{f^{13}} (7,2)$	$(1,2a8) \xrightarrow{f^{13}} (38,1)$

Table 8: Distribution of the trails for the different type of differentials in 13-round SIMECK32.

$\log_2 \Pr(Q)$	Type 1	Type 2	Type 3
-32	1	1	0
-33	0	0	0
-34	9	7	0
-35	6	5	0
-36	38	24	8
-37	44	28	2
-38	124	71	87
-39	166	96	79
-40	367	210	560
-41	521	308	868
-42	1014	625	2911
-43	1566	1002	5170
-44	2629	1752	12485
-45	4232	2975	22007
-46	6448	5101	43969
-47	9620	8234	75212
-48	13952	14439	133341
-49	19425	24653	220359
$\sum$	$2^{-27.88}$	$2^{-28.43}$	$2^{-27.29}$

Table 9: Number of trails and time to find them for the SIMECK48 differential $(400000, e00000) \xrightarrow{f^{20}} (400000, 200000)$.

$\log_2 \Pr(Q)$	#Trails	Pr(Differential)	T
-50	1	-50.0	3.72s
-51	0	-50.0	6.9s
-52	12	-48.0	19.78s
-53	6	-47.7520724866	31.77s
-54	80	-46.7145977811	42.62s
-55	68	-46.4301443917	55.68s
-56	413	-45.804012702	77.58s
-57	484	-45.5334136623	104.69s
-58	1791	-45.1367816524	180.02s
-59	2702	-44.8963843436	265.5s
-60	7225	-44.6271009401	528.39s
-61	12496	-44.4289288164	1068.95s
-62	28597	-44.2312406041	2603.59s
-63	52104	-44.0720542548	6146.77s
-64	111379	-43.9193398907	19276.9s
-65	207544	-43.7902765446	41938.08s
-66	238939	-43.7209043818	70720.98s
-67	228530	-43.6888725691	96657.81s
-68	229018	-43.6730860168	123706.38s
-69	276314	-43.6636455186	160688.8s
-70	271192	-43.6590352669	197354.41s
-71	269239	-43.6567522016	232641.34s
-72	267563	-43.6556191172	271083.28s
-73	266716	-43.6550547005	308072.68s
-74	227971	-43.6548135551	336027.17s

Table 10: Truncated differential for SIMECK32 obtained by extending $(8000, 4011) \xrightarrow{f^{13}} (4000, 0)$ in both directions until all bits are unknown.

Round	ΔL	ΔR	$*$	$*$
-4	***0**********	**********	15	16
-3	**000***0****1**	***0**********	11	15
-2	0*0000*000***01*	**000***0****1**	6	11
-1	0100000000010001	0*0000*000***01*	0	6
0	1000000000000000	0100000000010001	0	0
		13 rounds		
13	0100000000000000	0000000000000000	0	0
14	1*0000000000*000	0100000000000000	2	0
15	**00000*000**001	1*0000000000*000	5	2
16	***000**00***01*	**00000*000**001	9	5
17	***00***0*******	***000**00***01*	13	9
18	***0**********	***00***0*******	15	13
19	**********	***0**********	16	15

Table 11: Truncated differential for SIMECK64 obtained by extending $(0, 4400000) \xrightarrow{f^{26}} (8800000, 400000)$ in both directions until all bits are unknown.

Round	ΔL	ΔR	$*$	$*$
-8	**********0**********	**********	31	32
-7	**********0**00***0**********	**********0**********	28	31
-6	**********00*000**00***0*******	**********0**00***0**********	24	28
-5	**********0000000*000**00***0***	**********00*000**00***0*******	19	24
-4	*0****1***000000000000*000**00**	**********0000000*000**00***0***	13	19
-3	*00***01**00000000000000000*000*	*0****1***000000000000*000**00**	8	13
-2	*000**001*00000000000000000000	*00***01**0000000000000000*000*	4	8
-1	00000100010000000000000000000000	*000**001*0000000000000000000	0	4
0	00000000000000000000000000000000	00000100010000000000000000000000	0	0
		26 rounds		
26	00001000100000000000000000000000	00000000010000000000000000000000	0	0
27	000**001*10000000000000000000*	00001000100000000000000000000000	4	0
28	00***01***0000000000000000*000**	000**001*10000000000000000000*	9	4
29	0****1****00000000000*000**00***	00***01***0000000000000000*000**	14	9
30	**********000000*000**00***0****	0****1****00000000000*000**00***	20	14
31	**********0*000**00***0*********	**********000000*000**00***0****	25	20
32	***********00***0**************	**********0*000**00***0*********	29	25
33	***********0******************	***********00***0**************	31	29
34	*********************************	***********0******************	32	31